

CS474/674 Image Processing and Interpretation
Fall 2025 – Dr. George Bebis
Programming Assignment
Due Date: 12/08/2025

In this assignment, you will experiment with filtering in the frequency domain; the steps have been discussed in class extensively and are outlined on pages 312-313 in your textbook.

[40%] Experiment 1 (noise removal): The noisy image shown below (i.e., can be downloaded from course's webpage) has been generated by adding “cosine” (i.e., periodic) noise to the original image. **(a)** Using frequency domain filtering, use (i) a band-reject filter and (ii) a notch filter to remove the noise and show your results in each case. When presenting your results, make sure that you discuss how you chose the parameters of the filters and show the spectra for all images (before and after filtering) and filters. For comparison purposes, attempt to remove the noise in the spatial domain using Gaussian filtering (e.g., 7 x 7 or 15 x 15 filters). Compare your results between spatial and frequency domain methods. **(b)** Instead of removing the noise, devise a procedure using frequency domain filtering to extract the noise pattern and show your results. **(c)** Accurate and reliable face verification and recognition (see definitions below) relies heavily on high-quality face images. What are some potential consequences of failing to reduce noise or introducing artifacts due to algorithmic errors? This is a **free response question**; you can discuss both safety and economic implications. Provide proper citation for any information you obtain from other sources.



Definitions: Face verification is concerned with validating a claimed identity based on the image of a face (**one-to-one matching**). Face recognition is concerned with identifying a person based on the image of a face (**one-to-many matching**).

[20%] Experiment 2 (convolution in the frequency domain): Repeat the experiment described in Example 4.15 (page 317), using the “lenna” image. When presenting your results, make sure that you show the spectra for all images (before and after filtering) and filters. For comparison purposes, perform the convolution both in the spatial and frequency domains (i.e., see Figure 4.38, page 319) and compare your results.

[40%] Experiment 3 (homomorphic filtering): Many times, images suffer from shading problems due to uneven illumination. The role of homomorphic filtering is to alleviate such problems. In your experiments, use the image shown below (i.e., can be downloaded from the course's webpage). As discussed in class, the main idea behind homomorphic filtering is to separate the illumination and reflectance components by applying the logarithmic function on the image. You would then need to apply an appropriate high-pass filter, which will emphasize high frequencies and attenuate lower ones, preserving fine detail at the same time.



The high-pass filter to be used in your experiments is a **high-frequency emphasis** filter:

$$H(u, v) = (\gamma_H - \gamma_L) \left[1 - e^{-c[(u^2 + v^2)/D_0^2]} \right] + \gamma_L$$

where D_0 is the cutoff frequency of the filter and γ_L γ_H are the gains for the low and high frequencies correspondingly. Note that before you apply the filter on the image, you must first **center** it so that its center coincides with the center of the spectrum of the image.

Experiment with different parameter values. As a starting point, choose $D_0=1.8$, $c=1$, $\gamma_L=0.5$ and $\gamma_H = 1.5$. Then, keep the cutoff frequency the same and increase/decrease γ_L and γ_H . For example, assume combinations of γ_L and γ_H , with γ_L taking values from [0.0-1.0] and γ_H taking values from [1.0-2.0]. Show and comment on your results. Which set of parameters seems to be

working the best? Is there a consistency in your results as γ_L increases/decreases? What about γ_H ? When presenting your results, make sure that you show the spectra for all images (before and after filtering) and filters.

[20% Extra Credit] Experiment 4 (image restoration/motion blur): In this experiment, you will experiment with (i) Inverse filtering and (ii) Wiener filtering.

$$\hat{F}(u,v) = (1/H(u,v)) G(u,v) \quad \hat{F}(u,v) = \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \cdot \frac{G(u,v)}{H(u,v)}$$

Assume that $G(u,v) = H(u,v)F(u,v) + N(u,v)$ where $H(u,v) = \frac{T}{\pi(u\alpha + vb)} \sin(\pi(ua + vb)) e^{-j\pi(ua + vb)}$

First, apply the motion blur function $H(u,v)$ on the “lenna” image using $a=b=0.1$ and $T=1$ (i.e., an example is shown in Figure 5.26, page 404). Make sure that both $F(u,v)$ and $H(u,v)$ are centered to correctly degrade “lenna”. Then, add Gaussian noise to the blurred “lenna” image. Note that adding Gaussian noise to an image is not the same as convolving the image with a Gaussian filter (i.e., this will actually reduce image noise). You should be using the **Box-Muller** transformation to generate Gaussian noise (see link posted in the webpage of the course). To add Gaussian noise to an image, do the following for each pixel $I[i][j]$:

- (1) Call the function provided; suppose the value returned is $G(\mu, \sigma)$
- (2) $I_{noisy}[i][j] = I[i][j] + G(\mu, \sigma)$

Experiment with the following (μ, σ) values: $(0, 1)$, $(0, 10)$, and $(0, 100)$ (note: make sure that the final pixel values are in $[0, 255]$). For each degraded image, first experiment with Inverse filtering. Try different values for the “radius” parameter and show your results in the spirit of Figure 5.27 (page 406). Then, experiment with Wiener filtering. Try different “K” values (e.g., $0.0001, 0.001, 0.01$, and 0.1) and compare your results with Inverse filtering in the spirit of Figure 5.29 (page 410).